

Article

Grape Target Detection Method in Orchard Environment Based on Improved YOLOv7

Fuchun Sun ^{1,2,*}, Qiurong Lv ^{1,3}, Yuechao Bian ³ , Renwei He ^{4,5}, Dong Lv ^{5,6}, Leina Gao ¹, Haorong Wu ⁷ and Xiaoxiao Li ³

¹ Entrepreneurship College, Chengdu University, Chengdu 610106, China

² Development Planning Department, Chengdu University, Chengdu 610106, China

³ School of Mechanical Engineering, Chengdu University, Chengdu 610106, China

⁴ Engineering & Technical College, Chengdu University of Technology, Chengdu 610059, China

⁵ Chengdu Institute of Metrology Verification and Testing, Chengdu 610000, China

⁶ School of Mechanical and Electrical Engineering, Southwest Petroleum University, Chengdu 610500, China

⁷ School of Electronic Information and Electrical Engineering, Chengdu University, Chengdu 610106, China

* Correspondence: sfc@cdu.edu.cn; Tel.: +86-199-8040-9129

Abstract: In response to the poor detection performance of grapes in orchards caused by issues such as leaf occlusion and fruit overlap, this study proposes an improved grape detection method named YOLOv7-MCSF based on the You Only Look Once v7 (YOLOv7) framework. Firstly, the original backbone network is replaced with MobileOne to achieve a lightweight improvement of the model, thereby reducing the number of parameters. In addition, a Channel Attention (CA) module was added to the neck network to reduce interference from the orchard background and to accelerate the inference speed. Secondly, the SPPFCSPC pyramid pooling is embedded to enhance the speed of image feature fusion while maintaining a consistent receptive field. Finally, the Focal-EIoU loss function is employed to optimize the regression prediction boxes, accelerating their convergence and improving regression accuracy. The experimental results indicate that, compared to the original YOLOv7 model, the YOLOv7-MCSF model achieves a 26.9% reduction in weight, an increase in frame rate of 21.57 f/s, and improvements in precision, recall, and mAP of 2.4%, 1.8%, and 3.5%, respectively. The improved model can efficiently and in real-time identify grape clusters, providing technical support for the deployment of mobile devices and embedded grape detection systems in orchard environments.

Keywords: image processing; grape target detection; YOLOv7; lightweight network



Academic Editor: José Casanova Gascón

Received: 21 November 2024

Revised: 16 December 2024

Accepted: 26 December 2024

Published: 27 December 2024

Citation: Sun, F.; Lv, Q.; Bian, Y.; He, R.; Lv, D.; Gao, L.; Wu, H.; Li, X. Grape Target Detection Method in Orchard Environment Based on Improved YOLOv7. *Agronomy* **2025**, *15*, 42. <https://doi.org/10.3390/agronomy15010042>

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Grapes have high economic and food value and are widely cultivated all over the world. However, at present, grapes are mainly harvested manually, which is inefficient and has high labor costs [1]. Data show that the labor cost of manual grape harvesting accounts for 47.8% of the total expenditure, while the time consumed by manual harvesting accounts for 30% of the total working time [2,3]. With the continuous development of agricultural mechanization, the development of grape-picking robots has become an important development direction for the grape-growing industry [4]. How to quickly and accurately achieve the recognition of grape targets in complex environments such as light, shade, and overlap is the main problem facing the picking robot [5]. In traditional digital image processing, Xiong et al. [6] employed color models, an improved Chan-Vese model, and morphological processing methods to detect grape fruits, achieving a recognition rate of 91.67%. However,

color models are extremely sensitive to lighting conditions. Under different lighting conditions, the color of grape fruits can undergo significant changes, leading to a decline in recognition performance. Marquez et al. [7] utilized a method combining pixel classification and morphological operators to separate grape image pixels from the background and color components, resulting in an accuracy of only 84% on different datasets, which is relatively low. Miao et al. [8] proposed a two-step measurement strategy that analyzes overlapping grape contours through edge detection and contour fitting, achieving a recognition rate of 98.5%. However, contour-fitting algorithms may fail to accurately capture all details when dealing with complex shapes, especially for highly overlapping or irregularly shaped grapes, and tend to incur larger errors. Lv et al. [9] achieved recognition of green grapes in orchards using multi-source image fusion, feature extraction, compressed sensing, and segmentation techniques, reaching a recognition accuracy of 92.1%. However, the processes of multi-source image fusion and feature extraction generally require high computational resources, posing significant challenges to the performance of embedded devices in orchard environments. In summary, traditional digital image processing methods are greatly affected by factors such as lighting and occlusion from branches and leaves in the complex scenarios of orchards.

With the continuous improvement of computer computing power, deep learning algorithms have become a research focus in the field of object detection due to their strong robustness and efficiency [10]. Object detection models are mainly divided into two categories: two-stage object detection models based on candidate regions, with representative models such as Fast Region-based Convolutional Neural Networks (Fast R-CNN) [11] and Faster Region-based Convolutional Neural Networks (Faster R-CNN) [12]. Fast R-CNN first inputs an image and multiple Regions of Interest (RoIs) into a convolutional network, where each RoI is pooled into a fixed-size feature map. This is then mapped to a feature vector through fully connected layers and classified and localized using a Softmax layer and a bounding box regression layer. By introducing RoI pooling layers and sharing convolutional feature maps, this method enhances detection accuracy. In Faster R-CNN, the image is first input into the network, where feature maps are generated through convolutional operations. The convolutional features are then fed into a Region Proposal Network (RPN) to generate rectangular region proposals. A classifier is used to determine whether the proposed region matches a specific class; if it matches, the region box undergoes regression processing. This process achieves convolutional shared computation with the Fast R-CNN network, significantly increasing the model's accuracy. Li et al. [13] replaced VGG16 in the network with a multi-level network structure and introduced RoiAlign to replace RoiPooling, achieving a detection rate of 87.33% for strawberries. Tu et al. [14] developed an MS-F RCNN model, which builds upon the Faster R-CNN framework by incorporating additional feature extractors. The model was trained separately for color and depth detectors, achieving a detection rate of 93.1% for passion fruit. Wan et al. [15] improved the pooling and convolutional layers of the Faster R-CNN network, designing a multi-class fruit recognition algorithm with an average detection rate of 91%. Although two-stage detection models are efficient for detecting multiple object classes, they have the drawbacks of larger model weights and slower detection speeds.

The other class is a regression-based single-stage target detection model that omits the candidate region generation step, allowing for direct prediction and output of the category probability of the target. Its representative models include Single Shot MultiBox Detector (SSD) [16], You Only Look Once (YOLO) [17], and others. The SSD algorithm employs VGG-16 as a feature extraction network, replacing the final global pooling layer and fully connected layers with convolutional layers, and adding multiple convolutional layers for down-sampling to form a complete network structure. During sampling, SSD utilizes

prior boxes with multiple scales and aspect ratios to predict the bounding boxes of the objects. After extracting features through the convolutional neural network, the model ultimately performs classification and regression of the objects. The YOLO model primarily consists of three main stages: the input layer, backbone network, and head network. The input layer preprocesses the image through methods such as adjusting resolution, cropping the background, correcting distortions, and eliminating noise. The backbone network accurately identifies and distinguishes input images to catch target features of different sizes. The head network integrates image feature information using a feature pyramid structure, enhancing the representational capability of the target area features. Simultaneously, the model assigns category labels to the detected objects and performs predictions through confidence calculations. In the orchard detection environment, object detection models need to be deployed on embedded devices with lower performance and reduced computational load. In contrast, single-stage object detection models offer faster detection speeds and fewer model weights, making them more suitable for complex orchard environments. Agarwal et al. [18] proposed an improved SSD network utilizing a deep CNN structure based on Darknet-19 for the identification of mango fruits on trees, achieving an average recognition rate of 92.43%. Guo et al. [19] introduced a YOLOv4+ model that integrates a non-parametric attention mechanism into the Resblock-body of the backbone network while employing a focal loss function, resulting in an average recognition rate of 94.25% for grapes. Chen et al. [20] designed a lightweight model called GA-YOLO, utilizing the SE-CSPGhost backbone network and incorporating an adaptive spatial feature fusion mechanism. The proposed model achieved an average recognition rate of 96.87% for grape detection. However, both SSD and YOLO models demonstrate suboptimal performance in object detection within orchard environments characterized by dense fruit distribution, foliage occlusion, and uneven lighting. This calls for further research and improvements to enhance the accuracy of these models in achieving effective detection.

In summary, to address the challenges of grape recognition difficulty and extended detection times caused by dense fruits, occlusion, and overlapping in orchard environments, this study proposes a YOLOv7-MCSF network for grape target detection. This network enhances the structure of YOLOv7, achieving efficient recognition of grapes and providing essential support for the mechanized harvesting of grapes.

The main contributions of this study are as follows:

- (1) A lightweight MobileOne network is utilized as the backbone of the model. By incorporating depthwise convolutions from MobileOne, the model is optimized to reduce both the number of parameters and the computational complexity, making the object detection model more suitable for deployment on mobile devices with lower performance and reduced computational capacity.
- (2) A CA attention mechanism is integrated into the neck network, and the parallel computation structure of the pyramid pooling component in SPPCSPC is optimized to a serial computation structure. This adjustment enhances the capability of feature information fusion while improving the speed of image feature integration.
- (3) The Focal-EIoU loss function is employed to enhance the accuracy of object localization. The Focal-EIoU addresses the shortcomings of CIoU in dealing with object overlap situations and maintaining scale invariance, thereby improving the detection of misclassifications and missed detections in grape detection within orchard environments.

2. Materials and Methods

2.1. Collection of Datasets

The experimental data of this study were collected from a grape plantation in Chengdu, Sichuan Province, and the shooting period was 9:00–17:00 (sunny day). When acquiring

images of grape bunches on a sunny day, non-uniform lighting conditions tend to prevail in the morning and evening, and strong lighting conditions tend to prevail in the middle of the day. Grape fruit images under non-uniform illumination, due to the influence of clouds and the shooting device, some grape bunches (or the same bunches of different regions, the upper light is strong, the lower light is weak) are subjected to strong light irradiation, and the bunches of images have low local light intensity, which has a strong influence on grape target detection. An image of grapes under bright light consists of the whole domain irradiated by a uniformly strong light, but the light may change significantly at a certain moment.

To simulate the working conditions of the grape-picking machine vision system, images were taken from multiple angles and at various distances. The distance between the camera and the grape bunches ranged from 10 cm to 80 cm, with frontal shooting accounting for about 65% of the images and tilted-angle shooting supplementing it with approximately 35%. In order to ensure diversity in the test image samples, grape images were collected under various conditions, including single-target fruit, multi-target fruit, frontlight shooting, backlight shooting, uneven occlusion, leaf occlusion, and grape cluster occlusion. A total of 1300 original grape images containing 8265 bunch targets were obtained and saved in JPG format, with a maximum image resolution of 3024×4032 pixels. Figure 1 shows the images of grape samples acquired under different bunch configurations, lighting conditions, and shading scenarios.

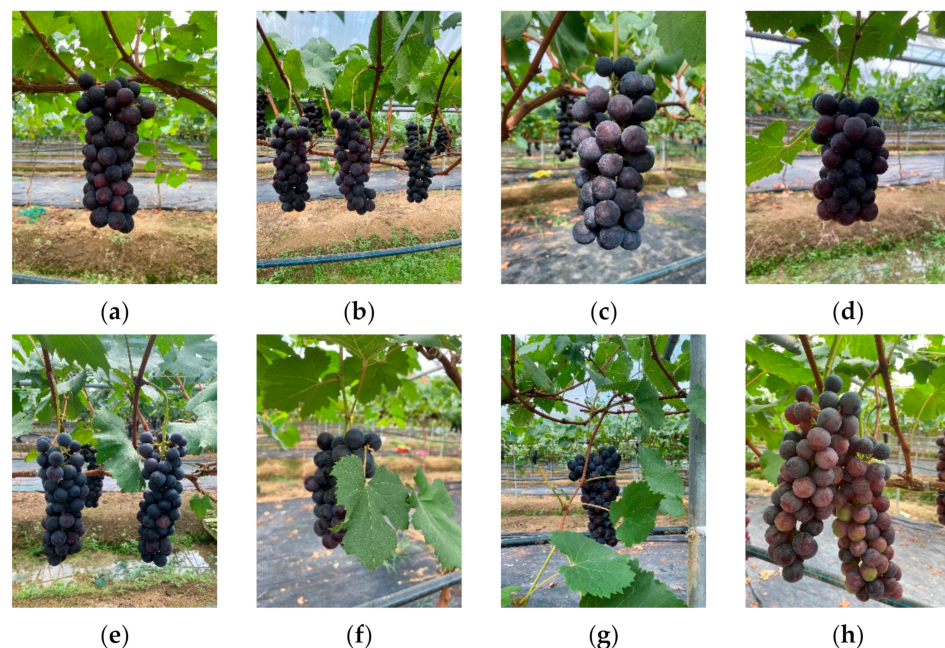


Figure 1. Grape sample images under different conditions. (a) Single target; (b) Multiple target; (c) Frontlight; (d) Backlight; (e) Uneven lighting; (f) Blade occlusion; (g) Stem occlusion; (h) Fruit overlap.

2.2. Production of Datasets

In this study, the grape dataset was manually annotated using Labelling (version 1.8.6), an image visualization annotation tool. The minimum bounding rectangle for the grape string was selected in the annotation frame, and the label was designated as “grape”. Due to the complex environment of the grape orchard, image augmentation techniques were applied to enhance the model’s generalization ability and avoid overfitting during training. These augmentation techniques included adjusting brightness, mirroring, flipping, random translation, adding Gaussian noise, and applying Gaussian filtering, as shown in Figure 2. As a result, the augmented dataset became four times larger than the original, yielding

5200 images. The augmented dataset was then divided in an 8:1:1 ratio, resulting in a training set of 4160 frames, a validation set of 520 frames, and a test set of 520 frames.

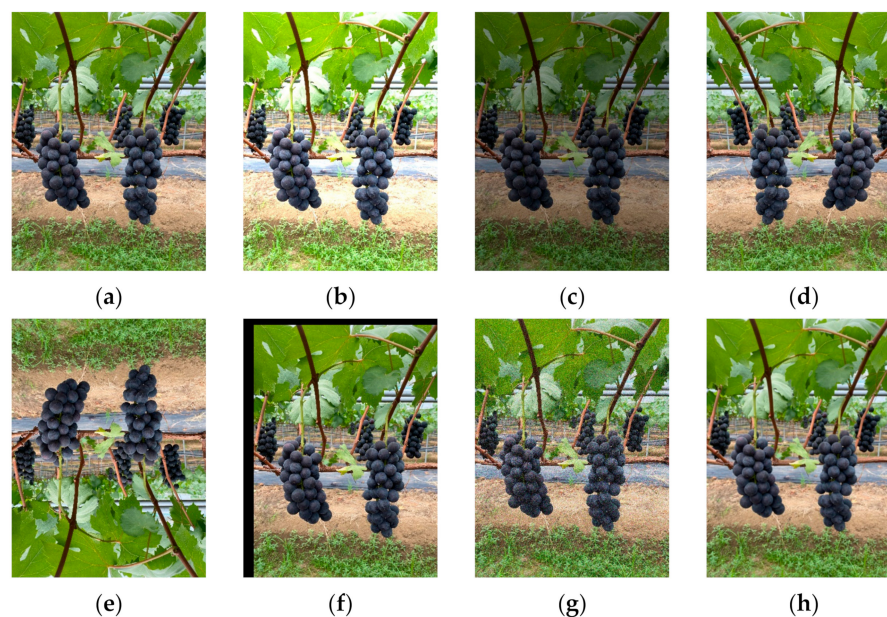


Figure 2. Data enhancement example. (a) Original image; (b) Increase brightness; (c) Reduced brightness; (d) Mirror image; (e) Vertical flip; (f) Random translation; (g) Gaussian noise; (h) Gaussian filtering.

2.3. Classic YOLOv7 Network Architecture

YOLOv7 [21] is a single-stage object detection network characterized by its rapid detection speed, high accuracy, and strong generalization capabilities. It demonstrates excellent flexibility and adjustability while effectively maintaining both efficiency and accuracy in the model. As shown in Figure 3, The YOLOv7 network structure consists of four parts: the input layer, the backbone feature extraction network module, the neck network module, and the output detection layer head network module. The input layer preprocesses the dataset through data enhancement to increase diversity and enhance the model's training. The backbone network mainly includes multiple convolutional layers, an ELAN module, and an MPConv module. The ELAN module and ELAN-W module, which include advanced features, are efficient network architectures in YOLOv7 and improve the model's target detection capability without increasing computation by adjusting the network's depth and adopting special connection methods. The MPConv module pools feature maps and then merges them with convolutionally processed feature maps, further optimizing the convolutional layers' ability to capture target features of different sizes. The neck network uses a feature pyramid structure to integrate shallow and deep feature information, enhancing the representation of features within the target area. Notably, the SPPCSPC module adds multiple parallel max-pooling (5×5 , 9×9 , 13×13) after the convolution process to obtain different sensory fields, thus enhancing the fusion ability of feature information. In the detection head, the operation of reconnection and channel merging of feature maps is performed through the Rep structure. Additionally, the residual design, which helps with training, is converted into a simple convolution in the actual prediction. This not only reduces the amount of network computation but also preserves the model's performance.

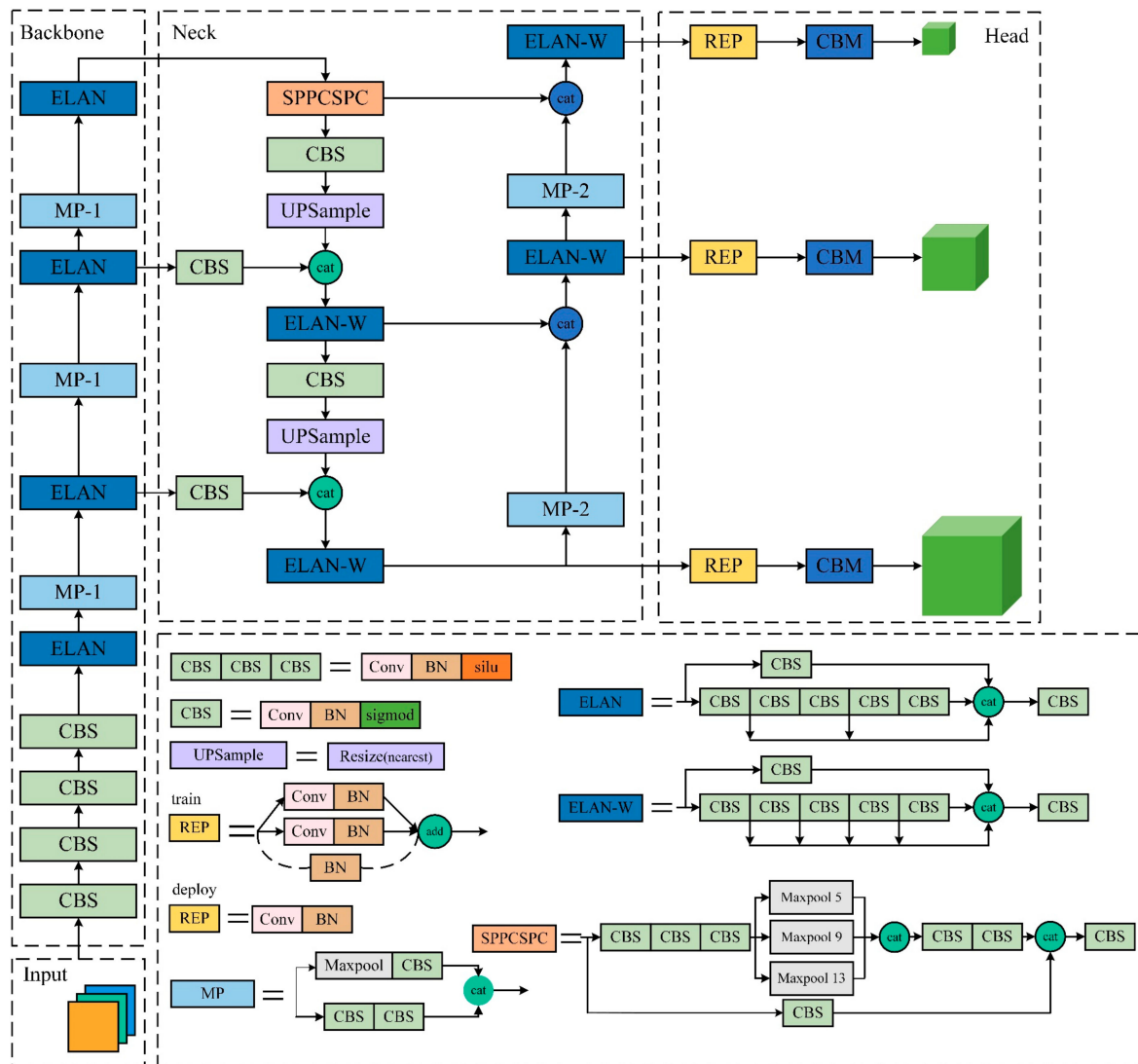


Figure 3. YOLOv7 network architecture.

2.4. YOLOv7-MCSF Grape Target Detection Model

2.4.1. Network Structure

In real orchard environments, grape clusters are dense, and the branches and leaves are lush, resulting in some grape fruits being obscured by leaves and branches, as well as overlapping with one another. This blockage leads to the concealment of critical characteristic information about the grapes, which hampers the performance of grape target detection [19]. To address this issue, this study proposes a new YOLOv7-MCSF grape target detection model, as shown in Figure 4. In the backbone network of YOLOv7, the ELAN module concatenates four features. However, since each channel contains different feature information, this direct concatenation can easily cause feature isolation among the various convolutional extractions, thereby degrading the model's detection performance in complex scenes. Therefore, a high-performance lightweight network, MobileOne, has been integrated into the YOLOv7 model, along with the incorporation of a Coordinate Attention (CA) module in the neck network. This enhancement allows the model to extract critical grape features more effectively and reduces the impact of background factors. Additionally, YOLOv7 utilizes the SPPCSPC module that employs multiple parallel max-pooling operations to obtain different receptive fields, thereby distinguishing images of varying sizes. However, due to the parallel pooling operations of the SPPCSPC convolutional

kernels, there is an increased computational burden on the network. Consequently, this study employs the SPPFCSPC module, replacing the three parallel max-pooling operations with serial max-pooling to improve the speed of image fusion. Finally, the Focal-EIoU loss function is utilized to enhance the accuracy of object localization, enabling the model to make more precise predictions.

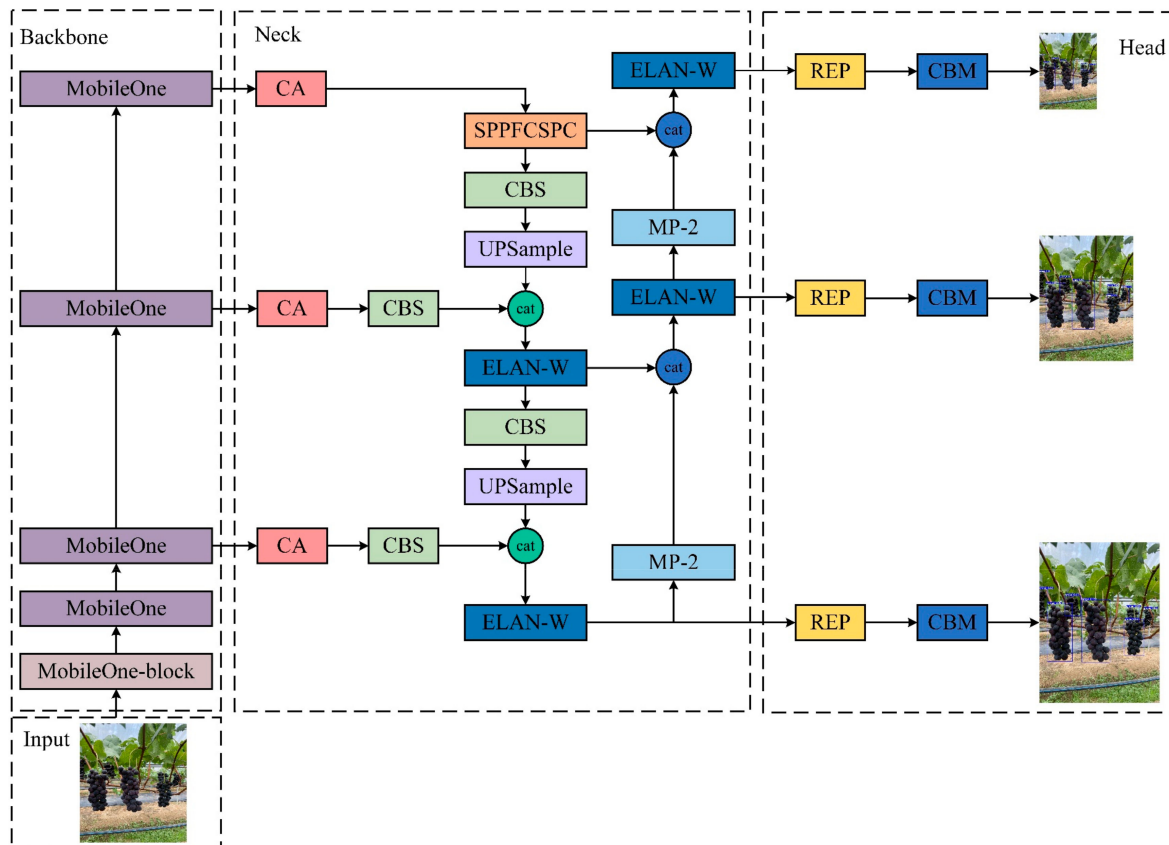


Figure 4. YOLOv7-MCSF network structure.

2.4.2. MobileOne Network

In the orchard detection environment, the model needs to be simplified to create a lightweight backbone network due to limited device performance. The MobileOne [22] network is a lightweight deep learning model built on the MobileNet architecture and the RepVGG convolutional neural network, which are both simple, efficient, and plug-and-play. MobileOne consists of numerous MobileOne Blocks that utilize depthwise separable convolution and reparameterization techniques to modify the network structure. Figure 5 shows the structure of the MobileOne Block, with the blocks for depthwise convolution and pointwise convolution used for network training on the left. The depthwise convolution module convolves each channel independently and merges the convolution results. In this setup, the leftmost branch is a 1×1 convolution, the middle branch is an over-parameterized 3×3 convolution, and the rightmost branch includes a jump connection containing Batch Normalization (BN) layers. The pointwise convolution consists of two branches of over-parameterized 1×1 convolution and a jump connection that includes BN layers to achieve the fusion of depthwise convolution results. During the inference stage, the reparameterization technique from RepVGG is applied to reduce extra computation by fusing the training results of the depthwise convolution part and the pointwise convolution part into a 3×3 depthwise convolution and a 1×1 pointwise convolution, respectively. This process results in a simplified model with a reduced number of parameters and faster

inference. Therefore, MobileOne is introduced as the backbone network to simplify the model and achieve lightweight operation.

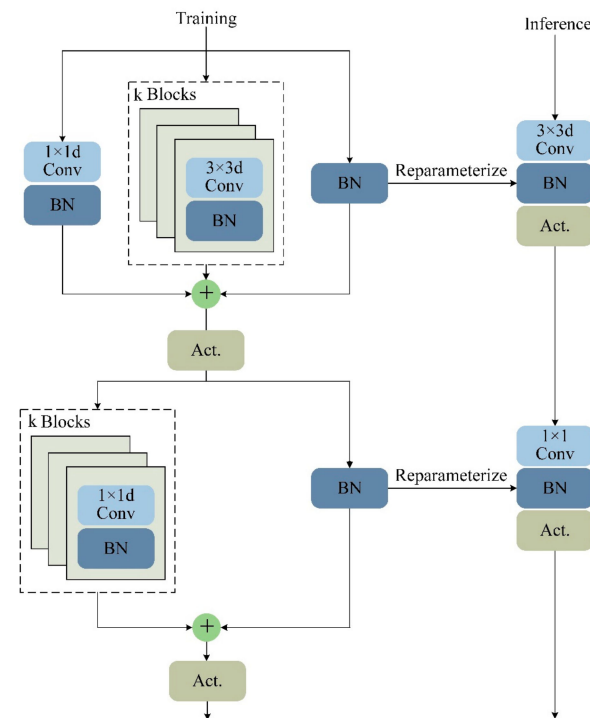


Figure 5. MobileOne Block structure.

2.4.3. CA Attention Mechanism

The improved MobileOne network achieves a lightweight model but lacks the ability for global feature extraction. On this basis, the attention mechanism module has been embedded into the model to enhance the YOLOv7-MCSF model's ability to recognize and utilize grape features. Commonly used lightweight attention mechanism modules include the Channel Attention Mechanism Module Squeeze-and-Excitation Networks (SE) [23] and the Convolutional Block Attention Mechanism Module Convolutional Block Attention Module (CBAM) [24]. However, the SE module only focuses on attention in the direction of channel features and cannot capture attention in the direction of spatial features. Although the CBAM module focuses on the image both spatially and channel-wise, it comes with higher computational complexity. Therefore, this study enhances the model's detection capability without significant additional computational cost by incorporating Coordinate Attention (CA) [25].

The CA attention mechanism is not only flexible and lightweight, but also avoids generating a large computational overhead, and its structure is shown in Figure 6. CA first pools the feature maps from the horizontal and vertical spatial directions to generate fused features in the horizontal and vertical spatial directions to avoid the failure to retain positional information due to single pooling. The dimensionality of the feature map is reduced using a convolution kernel and then processed by batch normalization and activation function, after which the feature map is divided into two feature maps for horizontal and vertical spatial directions.

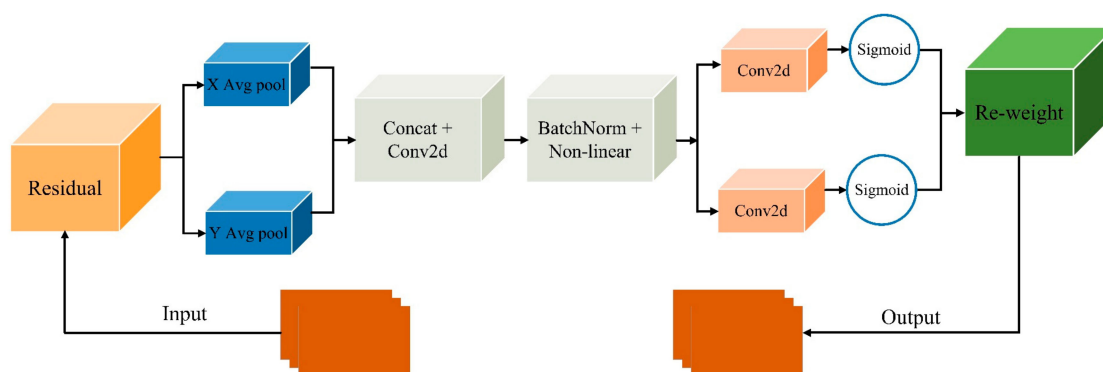


Figure 6. Structure of CA attention mechanism.

Secondly, the attention weights g^h and g^w of the feature map, corresponding to the direction of the height h and width w are calculated with the formula:

$$g^h = \sigma(F_h(f^h)), g^w = \sigma(F_w(f^w)) \quad (1)$$

where σ represents the nonlinear activation function, f is the spatial information feature map, f^h and f^w denote two independent tensors of the decomposition of f along the spatial direction, and F_h and F_w denote the feature maps obtained according to f .

Finally, a weighting operation is performed on the original feature map to obtain the weighted feature map of the CA module, which is given by the following formula:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \quad (2)$$

where $x_c(i, j)$ is the input on the c channel.

2.4.4. SPPFCSPC Module

YOLOv7 uses the SPPCSPC variant, which is improved by Spatial Pyramid Pooling (SPP) [26], as its feature fusion module. This module downsamples the feature map by using four different sizes of pooling kernels: 1, 5, 9, and 13. At the same time, it connects the bottom- and top-level features using Cross Stage Partial Channel (CSPC), which improves the generalization ability of the network by fusing the feature maps from four different sensing fields, achieving multi-scale feature fusion to adapt to detection objects of varying sizes and resolutions. However, the SPPCSPC module employs different-sized convolutional kernels for parallel pooling operations, which increases the computational effort of the network. Therefore, the SPP in the SPPCSPC module has been improved by changing its original parallel computing structure to a serial computing structure, resulting in the SPPFCSPC module. As shown in Figure 7, the SPPFCSPC performs spatial pyramid pooling and channel separation through feature maps. Concurrently, the three parallel spatial pooling convolution kernels in the SPPCSPC structure are replaced by serial spatial pooling convolution kernels of the same size. This change ensures that the feature maps passed to the next pooling layer have undergone processing by the previous pooling layer. Consequently, the ability to fuse different-scale feature information is enhanced and the speed of image feature fusion improves.

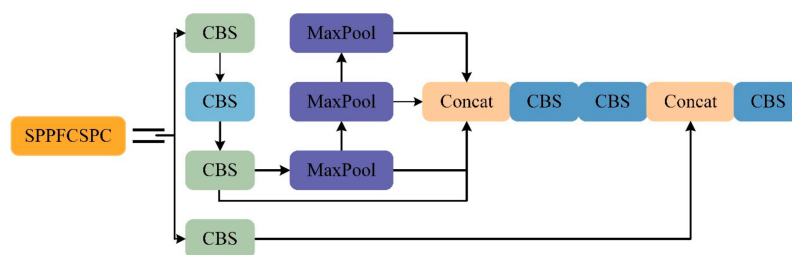


Figure 7. SPPFCSPC structure diagram.

2.4.5. Focal-EIoU Loss Function

The loss function plays a significant role in target detection, evaluating the discrepancy between model predictions and true labels, and guiding the optimization of model parameters. Choosing an appropriate loss function can effectively improve detection accuracy while promoting effective feature learning. The YOLOv7 model's bounding box employs the Complete Intersection over Union (CIoU) [27] loss function, which considers the distance between target centroids. This approach better handles targets with unconventional shapes and remains stable during training. Its calculation formula is as follows:

$$\begin{cases} Loss_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{C^2} + \alpha v \\ \alpha = \frac{v}{(1 - IoU) + v} \\ v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \end{cases} \quad (3)$$

where $\rho^2(b, b^{gt})$ denotes the centroid distance between the two; w, h , and b denote the width, height, and center point of the boundary of the prediction box, respectively; w^{gt}, h^{gt} , and b^{gt} denote the width, height, and center point of the boundary of the real bounding box, respectively; C is the diagonal length of the smallest enclosure of the two frames; α is the parameter of the equilibrium ratio; and v is used to measure the aspect ratio of the two. The computational structure of CIoU is shown in Figure 8.

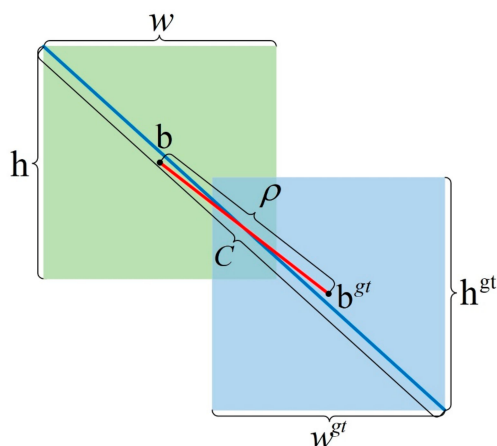


Figure 8. CIoU computational structure.

However, the aspect ratio in the CIoU formula describes a relative value that is somewhat fuzzy. This fuzziness cannot effectively address the overlapping situations between targets and maintain the scale invariance of the targets. As a result, the network's efficiency decreases, the convergence speed slows, and the regression results become less precise. In grape target detection, some clusters of grape fruits may be difficult to detect accurately due to lighting conditions, occlusion, overlap, and other factors. The model often ignores these challenging samples during detection, which weakens its ability to recognize such instances and leads to a stronger focus on easier samples. Consequently,

the model's training effectiveness may not meet expectations. To address these issues, this study introduces the Focal and Efficient Intersection over Union (Focal-EIoU) [28] as the boundary loss function. Focal-EIoU can enhance target location accuracy and increase focus on low-quality grape bunches, allowing the model to predict results more accurately. This is calculated using the following formula:

$$\begin{cases} Loss_{EIou} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{C^2} + \frac{\rho^2(w, w^{gt})}{C_w^2} + \frac{\rho^2(h, h^{gt})}{C_h^2} \\ Loss_{Focal-EIoU} = IoU^\gamma Loss_{EIou} \end{cases} \quad (4)$$

where the parameter γ is the hyperparameter used to control the curvature of the curve and C_w and C_h denote the width and height of the minimum enclosing box, respectively.

2.5. Experimental Environment

The experimental software environment consists of Windows 10 (64-bit), PyTorch 1.11.0, and Python 3.9. The hardware includes a Common KVM processor at 2.49 GHz with an octa-core architecture, 25 GB of RAM, and an NVIDIA GeForce RTX 4090 graphics card (NVIDIA, Santa Clara, CA, USA). It is also equipped with a parallel computing framework using CUDA 11.7 and cuDNN 8.4.1, which is adapted to the graphics card driver for GPU-accelerated model training. The original YOLOv7.pt pre-trained weights were selected, and the experimental parameters were set as outlined in Table 1.

Table 1. Experimental parameters.

Parameter Name	Set Alue
Image size	640 × 640
Multilinear process	16
Batch size	16
Weight attenuation coefficient	0.0005
Momentum parameter	0.937
Initial learning rate	0.01
Training round	300
Optimizer	SGD

2.6. Evaluation Indicators

In this study, Precision (P), Recall (R), mean Average Precision (mAP), model weights, and Frames per second (FPS) are used as the evaluation metrics of the model [29]. Where mAP0.5 denotes the mAP when IoU is 0.5.

2.6.1. Precision and Recall

Precision (P) is the proportion of samples that are true targets among those judged as such by the model. Recall (R) is the proportion of samples that are real targets that are correctly recognized as targets by the model. When evaluating the model performance, both metrics, P and R, should be considered, and their mathematical expressions are as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

where TP denotes the samples that were detected as grapes, FP denotes the samples that were incorrectly detected as grapes in the detection results, and FN denote the number of samples in which grape targets were detected as not grape targets.

However, there is a negative correlation between these two indicators; that is, improving accuracy often leads to a reduction in recall rates, and vice versa. Therefore, average accuracy is usually introduced for comprehensive analysis to assess the detection performance of the model regarding the target.

2.6.2. Average Precision and Mean Average Precision

The confidence level is used in target recognition to evaluate the probability that the current rectangular box is an object of some kind and takes a value between 0 and 1. When the confidence level is 1, it means that there is the greatest likelihood that the object in the rectangular box is of a certain class, and when the confidence level is 0, it means that there is the least likelihood. As shown below, by plotting the P-R curve, the area under the curve is calculated as the AP value.

$$AP = \int_0^1 P(R) dR \quad (7)$$

$$mAP = \frac{\sum_{i=1}^n AP_i}{n} \times 100\% \quad (8)$$

where mAP denotes the average value of AP in all categories and n denotes the total number of categories in which the target is recognized.

3. Results

3.1. Experimental Analysis of Attention Mechanisms

In order to exactly demonstrate the enhancement effect of the added CA attention module on the YOLOv7-MCSF detection model on the grape dataset, the three kinds of attention modules, SE, CBAM, and CA, were introduced into the same position of the YOLOv7-MCSF model, respectively. Then, the comparison experiments were carried out by comparing the models, and the results of the comparison are shown in Table 2.

Table 2. Comparative results of attention mechanisms.

Attention Mechanism	P/%	R/%	mAP _{0.5} /%	Model Weight/MB	FPS/(f·s ^{−1})
SE	90.7	88.3	95.5	52.0	75.19
CBAM	91.5	89.7	95.8	52.1	76.34
CA	92.8	91.5	96.2	52.1	79.37

By analyzing Table 2, we can conclude that the weight sizes of the models are essentially the same when different attention mechanisms are introduced. Compared to the introduction of Squeeze-and-Excitation (SE) and Convolutional Block Attention Module (CBAM), models incorporating the Channel Attention (CA) module show improvements of 2.1% and 1.3% in precision, 3.2% and 1.8% in recall, and 0.7% and 0.4% in mean average precision, respectively. Additionally, the detection speed improved by 4.18 frames per second (f/s) and 3.30 f/s, respectively.

To more intuitively observe the effect of feature extraction by introducing different attention mechanisms, this study uses Grad-CAM [30], a gradient-based class-activation thermal map, to carry out visualization operations. The visualization feature maps of introducing different attention mechanisms are shown in Figure 9. By observing the visualized feature maps, it was concluded that when the YOLOv7-MCSF model introduced the SE and CBAM attention mechanisms, the model had a weak extraction ability and a low response to the grape fruit features, whereas the two parallel feature encoding channels of the CA attention mechanism could reduce the background influence of the orchard

environment and enhance the model's attention to the grape target by encoding the channel relationship and a large range of effective information about the grapes. Therefore, the CA attention mechanism added in this study has good feasibility for grape target detection.

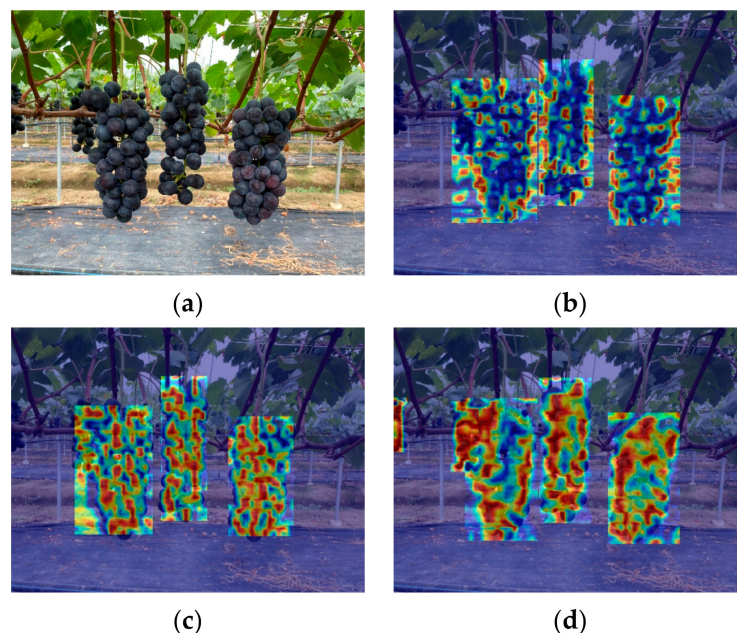


Figure 9. Characterization of different attention mechanisms. (a) Original figure; (b) SE attention mechanism; (c) CBAM attention mechanism; (d) CA attention mechanism.

3.2. Loss Function Comparison Experiment

To analyze the performance of different loss functions in grape detection, this experiment employs CIoU, DIoU, EIoU, and Focal-EIoU loss functions for the YOLOv7-MCSF model, respectively, and conducts a comparative analysis of the results. From the analysis of Figure 10, it is observed that when using the EIoU loss function, the model converges the slowest and has the highest loss value after convergence. When the CIoU and DIoU loss functions are applied, the convergence speed of the models is similar, and the loss values are lower than those of EIoU. In contrast, when the Focal-EIoU loss function is used, the model exhibits the fastest gradient descent during training, converging quickly within the first 30 iteration cycles and achieving the smallest loss value at convergence.

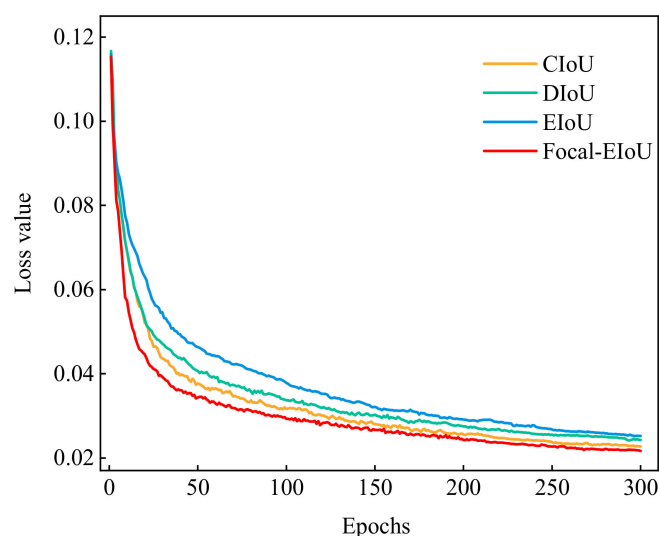


Figure 10. Iterative change loss curves for different loss functions.

3.3. Ablation Experiments

The ablation experiments were designed to analyze the value of each improvement. Using the homemade grape dataset and YOLOv7 as the baseline, the experiments were conducted by gradually introducing and combining different modules under the same experimental software, hardware, and hyperparameters. The experimental results are shown in Table 3.

Table 3. Ablation experiments with different modified methods.

Number	MobileOne	CA	SPPFCSPC	Focal-EIoU	mAP _{0.5} /%	Model Weight/MB	FPS/(f·s ^{−1})
1	×	×	×	×	92.7	71.3	57.80
2	✓	×	×	×	93.2	14.6	72.46
3	×	✓	×	×	94.1	65.6	58.48
4	×	×	✓	×	92.9	68.2	64.94
5	×	×	×	✓	93.5	71.3	62.11
6	✓	✓	×	×	95.1	32.5	68.96
7	×	✓	×	✓	94.5	64.7	63.70
8	✓	✓	✓	×	95.8	52.1	77.52
9	✓	✓	✓	✓	96.2	52.1	79.37

Note: “✓” means that the module has been added; “×” means that the module has not been added.

As can be concluded from Table 3, Experiment 2 uses the MobileOne backbone feature extraction network with a 0.5% reduction in model mAP_{0.5}, a 56.7 MB reduction in model weights, a 79.5% reduction in volume, and a 14.66 f/s increase in detection speed. The introduction of the MobileOne network resulted in a slight decrease in accuracy but improved FPS. Notably, this change significantly reduced the weights and computational load of the model. Experiments 3, 4, and 5 added the CA attention mechanism, SPPFCSPC, and Focal-EIoU to the YOLOv7 model, respectively; the mAP_{0.5} of the model was improved by 1.4%, 0.2%, and 0.8%, respectively; the model weights were reduced by 5.7 MB and 3.1 MB, respectively; and the detection speeds were improved by 0.68 f/s, 7.14 f/s, and 4.31 f/s. Experiment 6 uses the MobileOne backbone network and also introduces the CA attention mechanism, which improves the model mAP_{0.5} by 2.4%, reduces the model weight by 31.2 MB by 43.8%, and improves the detection speed by 11.16 f/s. Experiment 7 introduces CA and Focal-EIoU, which improves the mAP_{0.5} by 1.8%, reduces the model weight by 6.6 MB, and improves the detection speed by 5.9 f/s. Experiment 8 improves the MobileOne network, CA attention mechanism, and SPPFCSPC module by improving its mAP_{0.5} by 3.1%, model weight by 19.2 MB, and detection speed by 19.72 f/s. Experiment 9 shows the proposed YOLOv7-MCSF model, which achieves an mAP_{0.5} of 96.2%, a model weight of 52.1 MB, and a detection speed of 79.37 f/s. The experimental results obtained from the analysis conclusively demonstrate that the proposed network enhancement strategy effectively enhances the model performance.

3.4. Comparison Experiments Before and After YOLOv7 Improvement

In order to verify the effectiveness of the YOLOv7-MCSF model, we tested both the original YOLOv7 model and the YOLOv7-MCSF model under the same parameters and experimental conditions. The comparison of the performance metrics of the recognition before and after the improvement is shown in Table 4.

Table 4. Grape identification results for the pre- and post-improved YOLOv7 models.

Model	P/%	R/%	mAP _{0.5} /%	Model Weight/MB	FPS/(f·s ^{−1})
YOLOv7	90.4	89.7	92.7	71.3	57.80
YOLOv7-MCSF	92.8	91.5	96.2	52.1	79.37

Table 4 shows that the YOLOv7-MCSF model improves by 2.4%, 1.8%, and 3.5% in precision, recall, and mAP_{0.5}, respectively, compared to the YOLOv7 model. The YOLOv7-MCSF model recognizes grape targets more accurately and provides more reliable technical support for grape detection in orchard environments. Meanwhile, the YOLOv7-MCSF network model uses the MobileOne backbone network structure, and the model weight is reduced from 71.3 MB to 52.1 MB, which is a 26.9% reduction, and the detection speed is increased from 57.80 f/s to 79.37 f/s, which is a 21.57 f/s improvement.

After analyzing the experimental data, to further observe the comprehensive recognition effect of the original YOLOv7 model and the YOLOv7-MCSF model on grapes in complex environments in the orchard, some samples of the typical test set were selected for the comparison experiments in this experiment. As shown in Table 5, both models accurately recognize single-target and multi-target grapes when they are detected, but the YOLOv7-MCSF model performs even better with higher detection confidence. When the grapes were detected in downlight, backlight, uneven light, blade occlusion, and stem occlusion, the YOLOv7 model showed leakage, while the YOLOv7-MCSF model had less leakage and was more capable of detecting the grapes. In summary, YOLOv7-MCSF has higher recognition accuracy for grape bunches, as well as improved detection ability under light and occlusion conditions.

Table 5. Recognition effect of the model before and after improvement.

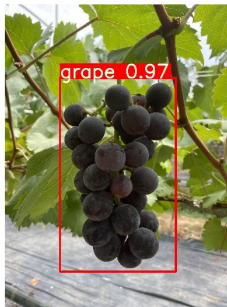
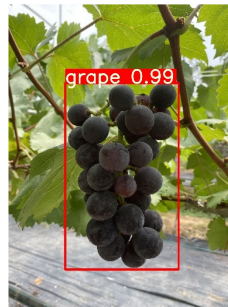
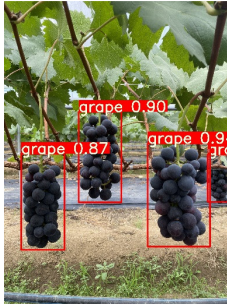
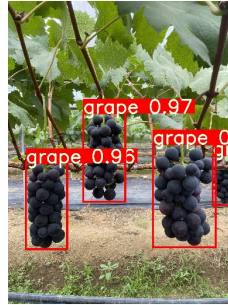
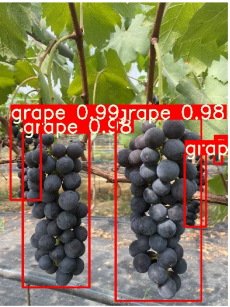
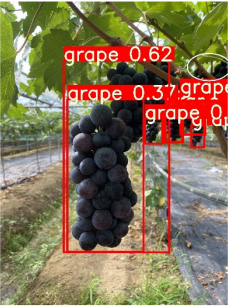
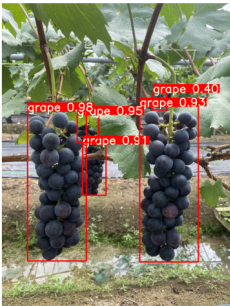
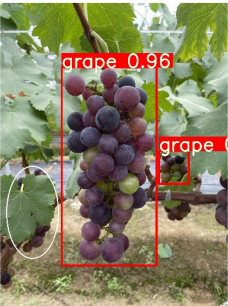
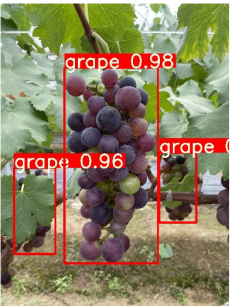
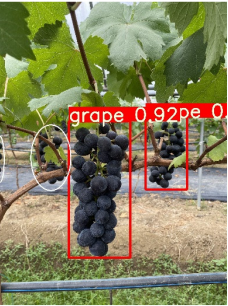
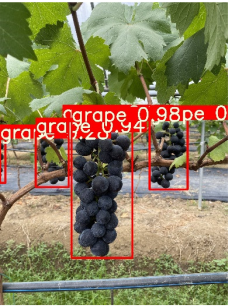
	Original Image	YOLOv7	YOLOv7-MCSF
Single target			
Multiple target			

Table 5. Cont.

	Original Image	YOLOv7	YOLOv7-MCSF
Frontlight			
Backlight			
Uneven light			
Blade occlusion			
Stem occlusion			

Note: The white circle box is the grape cluster that the original YOLOv7 model missed in detection.

3.5. Comparative Experiments with Different Models

To comprehensively evaluate the effectiveness of the YOLOv7-MCSF model in recognizing grape clusters, this study selects several classical models for comparative exper-

iments, including Faster R-CNN, SSD, YOLOv3 [31], YOLOv5 [32], YOLOv6 [33], and YOLOv8 [34], as well as some improved algorithms from other literature, such as YOLO-Grape [5], ESP-YOLO [35], and GDN [36]. The experimental results are presented in Table 6 and Figure 11.

Table 6. Comparison of different model detection.

Model	P/%	R/%	mAP _{0.5} /%	Model Weight/MB	FPS/(f·s ^{−1})
Faster R-CNN	75.2	73.6	79.4	108.0	28.65
SSD	85.1	83.5	87.8	90.6	72.99
YOLOv3	88.2	87.4	90.6	235.0	87.72
YOLOv5	89.7	88.6	91.5	88.5	69.93
YOLOv6	88.6	87.1	91.8	62.6	71.36
YOLOv8	90.1	88.5	93.7	43.7	91.42
YOLO-Grape	92.2	88.8	90.5	30.0	81.03
ESP-YOLO	92.7	82.3	89.4	11.2	11.11
GDN	91.8	88.5	93.7	24.9	0.96
YOLOv7-MCSF	92.8	91.5	96.2	52.1	79.37

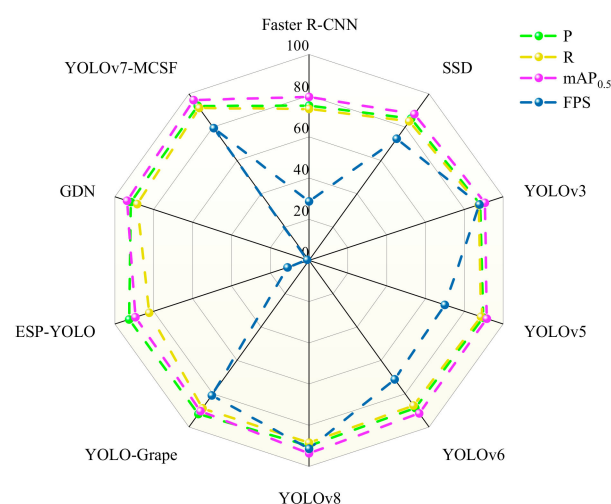


Figure 11. Comparison of accuracy of different models.

According to the experiments comparing different models presented in Table 6 and Figure 11, the YOLOv7-MCSF model demonstrates an accuracy rate of 92.8%. This is an improvement of 17.6%, 7.7%, 4.6%, 3.1%, 4.2%, 2.7%, 0.1%, 0.6%, and 1% over the Faster R-CNN, SSD, YOLOv3, YOLOv5, YOLOv6, YOLOv8, ESP-YOLO, YOLO-Grape, and GDN models, respectively. Additionally, the recall of the YOLOv7-MCSF model is 91.5%, which exceeds the other eight models by 17.9%, 8%, 4.1%, 2.9%, 4.4%, 3%, 2.7%, 9.2%, and 3%, respectively. The model also achieves a mean Average Precision at IoU 0.5 (mAP_{0.5}) of 96.2%, outperforming the previous models by 16.8%, 8.4%, 5.6%, 4.7%, 4.4%, 2.5%, 5.7%, 6.8%, and 2.5%. Regarding model size, the YOLOv7-MCSF model weighs 52.1 MB, which is a reduction of 55.9 MB and 38.5 MB compared to five classical models, respectively, such as Faster R-CNN and SSD. The detection speed of YOLOv7-MCSF is 79.37 frames per second (f/s), which is lower than YOLOv3, YOLOv8, and YOLO-Grape by 8.35 f/s, 12.05 f/s, and 1.66 f/s, respectively. However, it surpasses the detection speeds of Faster R-CNN, SSD, YOLOv5, YOLOv6, ESP-YOLO, and GDN models by 50.72 f/s, 6.38 f/s, 9.44 f/s, 8.01 f/s, 68.26 f/s, and 78.41 f/s, respectively.

In summary, the YOLOv7-MCSF model shows superiority in several indexes, has good comprehensive performance, has better real-time performance on mobile terminals, and is suitable for the detection of grape berries under actual orchards.

3.6. Comparison of Different Occlusion Types Detection

In order to evaluate the effectiveness of the model in detecting grape targets under different occlusion conditions, this experiment comprehensively considered a variety of specific occlusion situations that the robot may encounter during the picking process. Three types of occlusion were selected: stem occlusion, blade occlusion, and fruit overlap. The YOLOv7-MCSF model was used to test its detection performance under these three different occlusion categories. The experimental results are shown in Table 7.

Table 7. Detection results of different shading types.

Model	Type	P/%	R/%	mAP _{0.5} /%	FPS/(f·s ^{−1})
YOLOv7-MCSF	Stem occlusion	92.5	91.1	95.9	78.26
	Blade occlusion	91.2	90.5	94.3	77.58
	Fruit overlap	91.4	91.0	95.1	77.83

As shown in Table 7, the accuracy, recall, and mAP were 92.5%, 91.1%, and 95.9% under the stem occlusion condition; 91.2%, 90.5%, and 94.3% under the blade occlusion condition; and 91.4%, 91.0%, and 95.1% under the fruit overlap condition. The FPS for the three occlusion cases were 78.26 f/s, 77.58 f/s, and 77.83 f/s, respectively. Figure 12 demonstrates the grape target detection images under different occlusion conditions. As seen in Figure 12, the YOLOv7-MCSF network model is still able to accurately detect the grape fruits under branch and leaf occlusions as well as fruit overlaps, demonstrating strong robustness.

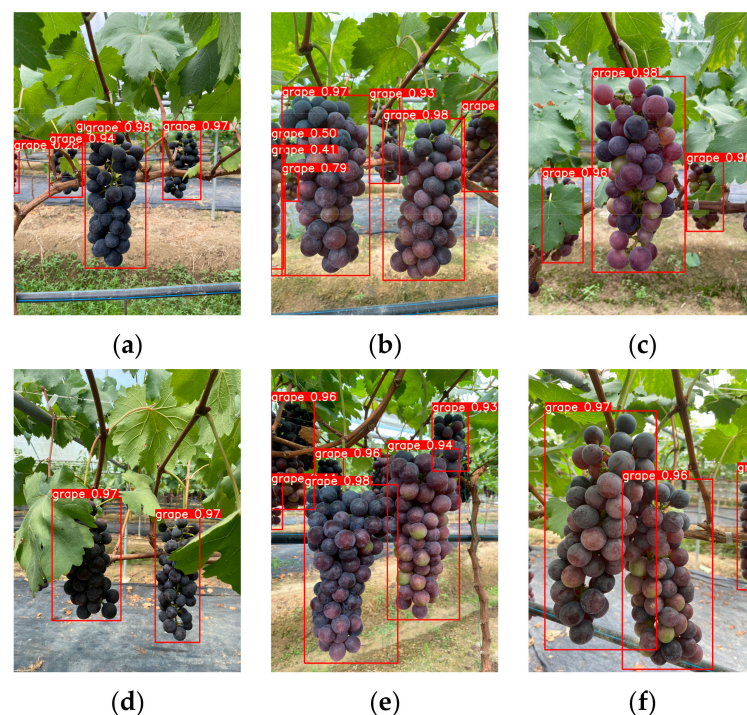


Figure 12. Schematic diagram of different occlusion types detection. (a,b) Stem occlusion; (c,d) Blade occlusion; (e,f) Fruit overlap.

4. Discussion

In the real orchard environment, the grape berries are obscured by leaves and branches, as well as the presence of mutual overlapping, which leads to the obscuring of the key feature information of the grapes and affects the effectiveness of target detection. Traditional digital image processing methods perform poorly under the influence of factors such as

light changes and branch and leaf occlusion. Deep learning algorithms are more efficient at detecting multi-class objects, but they face issues with larger model weights and slower detection speeds.

To address these challenges, this paper proposes a grapevine target detection method, YOLOv7-MCSF, which improves upon YOLOv7. The method significantly enhances the original YOLOv7 model by introducing the MobileOne, CA, and SPPFCSPC modules, as well as the Focal-EIoU loss function for better model performance. The experimental results indicate that the weight of the YOLOv7-MCSF model is reduced by 26.9%, and the frame rate has improved by 21.57 frames per second compared to the original YOLOv7 model. Simultaneously, its precision, recall, and mAP are improved by 2.4%, 1.8%, and 3.5%, respectively. These enhancements enable YOLOv7-MCSF to perform better in grape target detection within complex orchard environments.

It should be pointed out that the accuracy and efficiency of this method in recognizing grapes under different light, branch and leaf shading, etc. have been improved, but there are still a small number of missed detections. Based on this, subsequent plans will be made to upgrade the experimental environment to collect more raw data and enrich the dataset, as well as to investigate the combination of multi-task learning, edge computing, and the model to further enhance the applicability and reliability of the model.

5. Conclusions

In this study, a grape target detection method, YOLOv7-MCSF, based on YOLOv7, was proposed, which can effectively cope with complex conditions such as different lighting and branch and leaf occlusion in orchard environments and achieve accurate identification of grape fruits. In YOLOv7, the lightweight MobileOne backbone network is used to reduce the number of model parameters, and through experimental analysis, the CA attention mechanism, which has the most significant effect on model performance enhancement among SE, CBAM, and CA, is introduced to strengthen the ability to capture information about the key areas of grapes, and at the same time, the parallel computation structure of the pyramid pooling part of SPPCSPC is optimized and improved to a serial computation structure to enhance the fusion ability of feature information, and finally the Focal-EIoU loss function is used to improve the target location accuracy and improve the misdetection and omission problem of grape recognition in the orchard environment. Each improvement module improved the mAP_{0.5} of the YOLOv7-MCSF model by 0.5%, 1.4%, 0.2%, and 0.8%, respectively. The YOLOv7-MCSF model improves the precision, recall, and mAP_{0.5} of the model by 2.4%, 1.8%, and 3.5%, respectively, and the detection speed by 21.57 f/s compared with the original YOLOv7 by conducting experiments on the grape dataset. Compared with Faster R-CNN, SSD, YOLOv3, YOLOv5, YOLOv6, YOLOv8, YOLO-Grape, ESP-YOLO, and GDN models, its mAP_{0.5} was improved by 16.8%, 8.4%, 5.6%, 4.7%, 4.4%, 2.5%, 5.7%, 6.8%, and 2.5%, and was able to provide effective support for grapevine berry detection in orchard environments.

Author Contributions: Conceptualization, F.S. and Q.L.; methodology, Q.L., Y.B. and F.S.; formal analysis and investigation, R.H., D.L., L.G. and X.L.; Writing—original draft preparation, Q.L. and H.W.; Writing—review and editing, Q.L., Y.B. and H.W.; project administration, R.H. and X.L.; funding acquisition, F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Regional Innovation Cooperation Program of Sichuan Province (2024YFHZ0147), the National Student Entrepreneurship Practice Program of Sichuan Province 2024 (202411079004S), and the Special Classroom for Innovation and Entrepreneurship Education of CC National Crowd Space of Chengdu University 2024 (ccyg202401008).

Data Availability Statement: The authors will provide the raw data that supports the findings of this article upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, X.; Zou, X.; Tang, W.; Yan, Z.; Meng, H.; Luo, X. Unstructured road extraction and roadside fruit recognition in grape orchards based on a synchronous detection algorithm. *Front. Plant Sci.* **2023**, *14*, 1103276. [[CrossRef](#)] [[PubMed](#)]
2. Seyyedhasani, H.; Peng, C.; Jang, W.J.; Vougioukas, S.G. Collaboration of human pickers and crop-transporting robots during harvesting—Part I: Model and simulator development. *Comput. Electron. Agric.* **2020**, *172*, 105324. [[CrossRef](#)]
3. Jobbágy, J.; Dočkalík, M.; Krištof, K.; Burg, P. Mechanized grape harvest efficiency. *Appl. Sci.* **2021**, *11*, 4621. [[CrossRef](#)]
4. Wang, Z.; Xun, Y.; Wang, Y.; Yang, Q. Review of smart robots for fruit and vegetable picking in agriculture. *Int. J. Agric. Biol. Eng.* **2022**, *15*, 33–54.
5. Li, H.; Li, C.; Li, G.; Chen, L. A real-time table grape detection method based on improved YOLOv4-tiny network in complex background. *Biosyst. Eng.* **2021**, *212*, 347–359. [[CrossRef](#)]
6. Xiong, J.; Liu, Z.; Lin, R.; Bu, R.; He, Z.; Yang, Z.; Liang, C. Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (CCD) vision sensor with artificial illumination. *Sensors* **2018**, *18*, 969. [[CrossRef](#)]
7. Gonzalez-Marquez, M.R.; Brizuela, C.A.; Martinez-Rosas, M.E.; Cervantes, H. Grape bunch detection using a pixel-wise classification in image processing. In Proceedings of the 2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 4–6 November 2020; Volume 4, pp. 1–6.
8. Miao, Y.; Huang, L.; Zhang, S. A Two-Step Phenotypic Parameter Measurement Strategy for Overlapped Grapes under Different Light Conditions. *Sensors* **2021**, *21*, 4532. [[CrossRef](#)]
9. Lv, J.; Lv, X.; Ma, Z. A fruit recognition method of green grape images in the orchard. *N. Z. J. Crop Hortic. Sci.* **2022**, *50*, 1–16. [[CrossRef](#)]
10. Wu, H.; Luo, Z.; Sun, F.; Li, X.; Zhao, Y. An Improvement Method for Improving the Surface Defect Detection of Industrial Products Based on Contour Matching Algorithms. *Sensors* **2024**, *24*, 3932. [[CrossRef](#)]
11. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [[CrossRef](#)]
13. Li, J.; Zhu, Z.; Liu, H.; Su, Y.; Deng, L. Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R-CNN. *Ecol. Inform.* **2023**, *77*, 102210. [[CrossRef](#)]
14. Tu, S.; Pang, J.; Liu, H.; Zhuang, N.; Chen, Y.; Zheng, C. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. *Precis. Agric.* **2020**, *21*, 1072–1091. [[CrossRef](#)]
15. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2020**, *168*, 107036. [[CrossRef](#)]
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016, 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
18. Agarwal, D.; Bhargava, A. On-tree fruit detection system using Darknet-19 based SSD network. *J. Food Meas. Charact.* **2024**, *18*, 7067–7076. [[CrossRef](#)]
19. Guo, C.; Zheng, S.; Cheng, G.; Zhang, Y.; Ding, J. An improved YOLO v4 used for grape detection in unstructured environment. *Front. Plant Sci.* **2023**, *14*, 1209910. [[CrossRef](#)]
20. Chen, J.; Ma, A.; Huang, L.; Su, Y.; Li, W.; Zhang, H.; Wang, Z. GA-YOLO: A lightweight YOLO model for dense and occluded grape target detection. *Horticulturae* **2023**, *9*, 443. [[CrossRef](#)]
21. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
22. Vasu, P.K.A.; Gabriel, J.; Zhu, J.; Tuzel, O.; Ranjan, A. Mobileone: An improved one millisecond mobile backbone. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7907–7917.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.

24. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
25. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 13713–13722.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)]
27. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
28. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [[CrossRef](#)]
29. Zhou, J.; Zhang, Y.; Wang, J. RDE-YOLOv7: An improved model based on YOLOv7 for better performance in detecting dragon fruits. *Agronomy* **2023**, *13*, 1042. [[CrossRef](#)]
30. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [[CrossRef](#)]
31. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. In *Computer Vision and Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 1804, pp. 1–6.
32. Malta, A.; Mendes, M.; Farinha, T. Augmented reality maintenance assistant using yolov5. *Appl. Sci.* **2021**, *11*, 4758. [[CrossRef](#)]
33. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
34. Varghese, R.; Sambath, M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 18–19 April 2024; pp. 1–6.
35. Chen, J.; Chen, H.; Xu, F.; Lin, M.; Zhang, D.; Zhang, L. Real-time detection of mature table grapes using ESP-YOLO network on embedded platforms. *Biosyst. Eng.* **2024**, *246*, 122–134. [[CrossRef](#)]
36. Wang, W.; Shi, Y.; Liu, W.; Che, Z. An Unstructured Orchard Grape Detection Method Utilizing YOLOv5s. *Agriculture* **2024**, *14*, 262. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.